**LL(1)语法分析器实验报告**

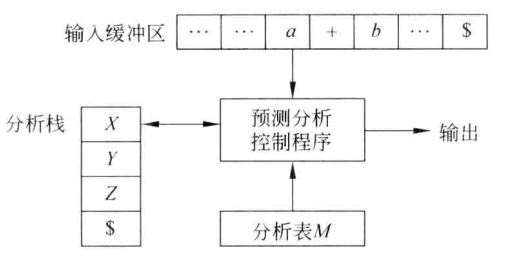
**付彬 2017211307班 2017211357**

1. **运行/开发环境及版本号**

OS：Windows 10

IDE：Visual Studio 2019

1. **LL(1)语法分析器设计**
2. **语法分析器模型**

****

1. **语法分析器模型具体实现**
2. **输入缓冲区**

****

1. **分析栈**

****

1. **预测分析控制程序**

****

1. **分析表**

****

1. **文法相关信息**
2. **原始文法**

**E -> E+T | E-T | T**

**T -> T\*F | T/F | F**

**F -> (E) | num**

1. **消除左递归后的文法**

**E -> TA**

**A -> +TA | -TA | ε**

**T -> FB**

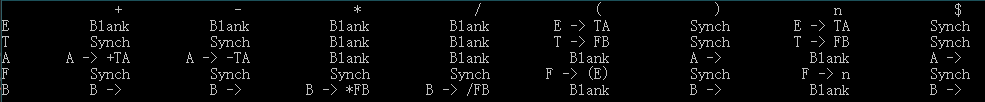
**B -> \*FB | /FB | ε**

**F -> (E) | num**

1. **First集和Follow集**

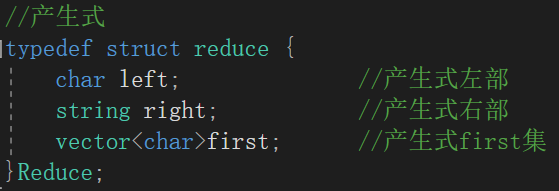
|  |  |  |
| --- | --- | --- |
|  | first | follow |
| E | ( , num | $ , ) |
| A | + , - , ε | $ , ) |
| T | ( , num | $ , ) , + , - |
| B | \* , / , ε | $ , ) , + , - |
| F | ( , num | $ , ) , + , - , \* , / |

1. **分析表（由程序打印得）**



1. **实现细节**
2. **数据结构**
3. **产生式**

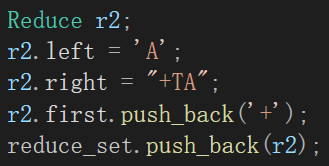
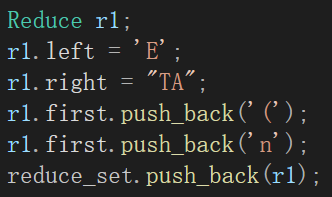
**产生式的结构体定义：**

****

**用vector作为产生式集合的容器：**

****

**将产生式初始化置入vector：**

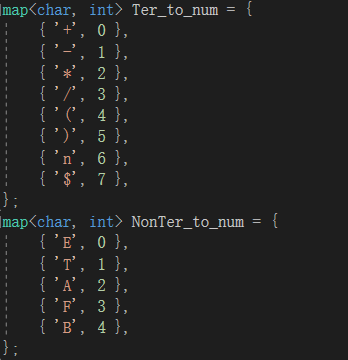


1. **终结符和非终结符**

**文法符号的存储：**

****

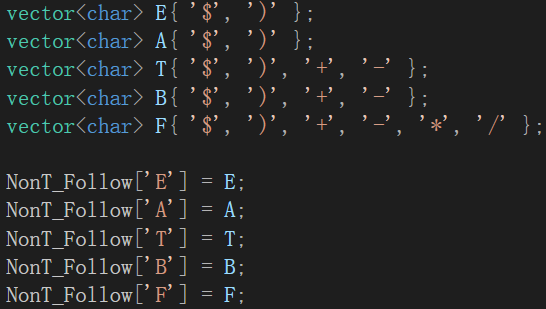
**实现了文法符号到数字的映射，在编程时，利于按文法符号的索引访问其他数据结构：**

****

1. **非终结符Follow集**

**使用map存储：**

****

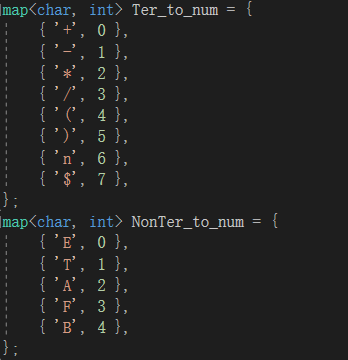
**初始化：  
**

1. **预测分析表**

**分析表存储：**

****

**访问预测分析表时，利用提到的文法符号到数字的映射填入矩阵的行下标和列下标，分析表在构建时也是按照下图终结符号和非终结符号的顺序填入产生式的。**

****

1. **分析栈**

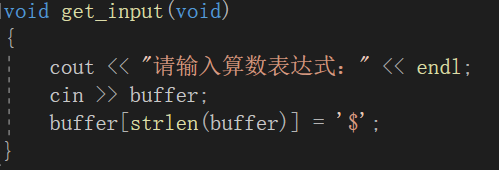
**分析栈存储：**

****

1. **输入缓存区**

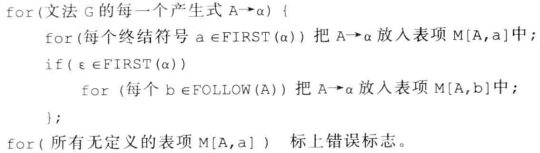
****

**在输入字符串的末尾添加$符号：**

****

1. **主要功能**
2. **构造语法分析表**

**书中算法：**

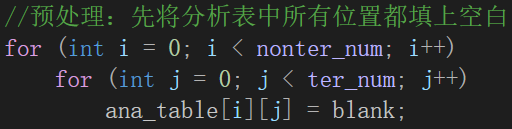
****

**本程序在实现书中算法的基础上，还增添了附加功能：**

1. **预先将分析表填满空白**

****

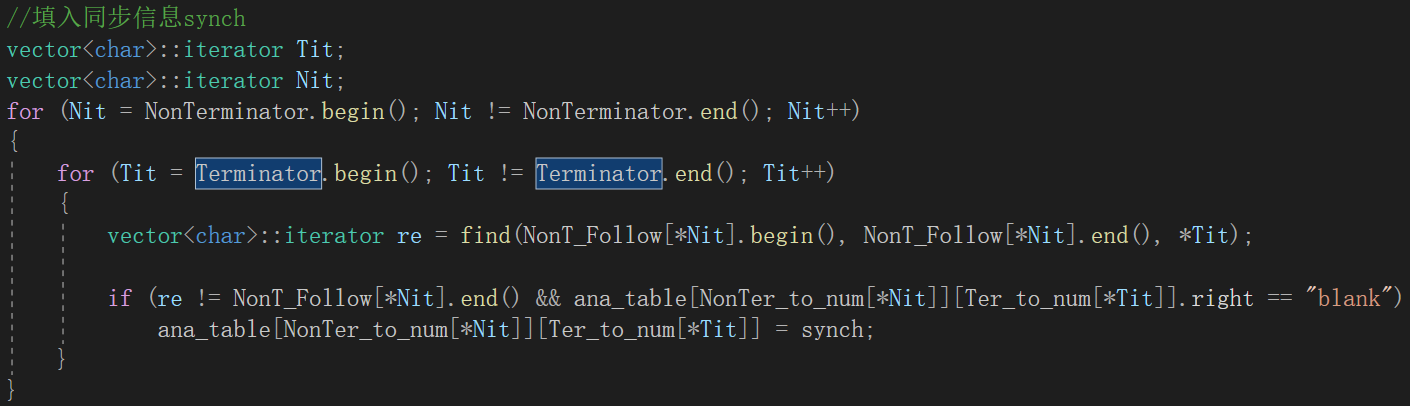
****

****

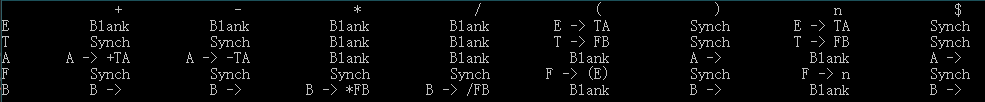
1. **填入同步信息，方便语法分析**

****

****

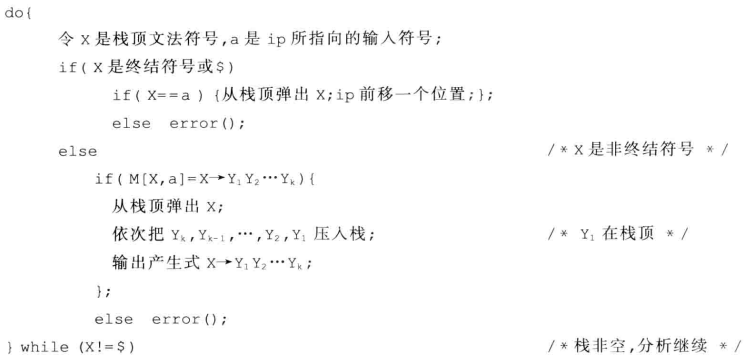


1. **最终结果**



1. **对给定文法进行语法分析**

**书中算法：**

****

**本程序在实现书中算法的基础上，还增添了附加功能：主要是丰富上图中error()函数中的内容。增添了更多错误处理功能，见下面。**

1. **检测语法错误，且能够报告出一些语法错误，用于提示用户更正**

**错误处理方式：**

1. **栈顶为终结符，且与输入缓存区起始符号不匹配：**

**将栈顶符号弹出**

1. **栈顶为非终结符，且分析表内容为blank**

**前移输入缓冲区指针**

1. **栈顶为非终结符，且分析表内容为synch**

**弹出栈顶符号**

**错误类别：**

**设置了下面4类错误：**

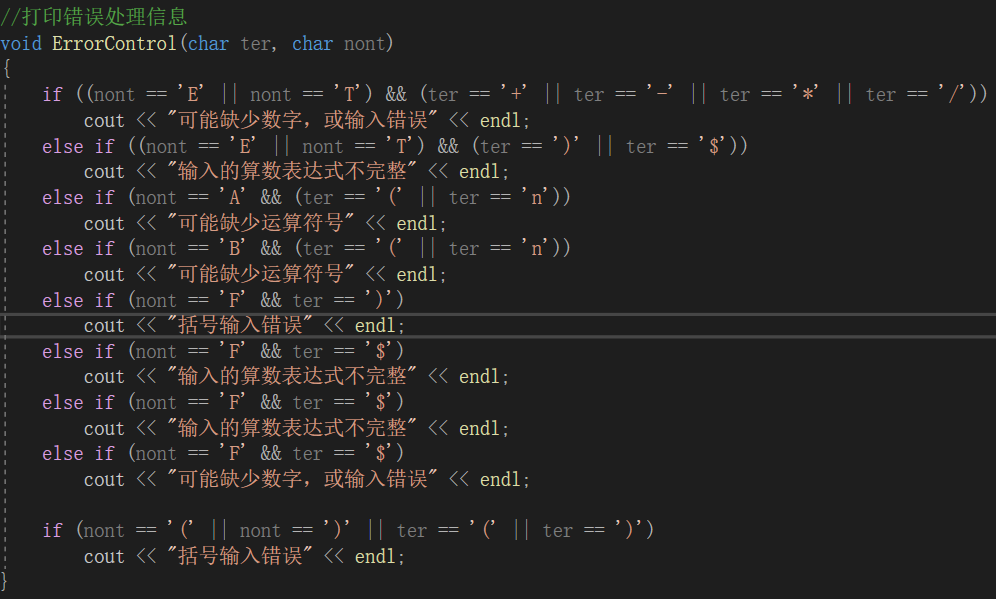
可能缺少数字，或输入错误

输入的算数表达式不完整

可能缺少运算符号

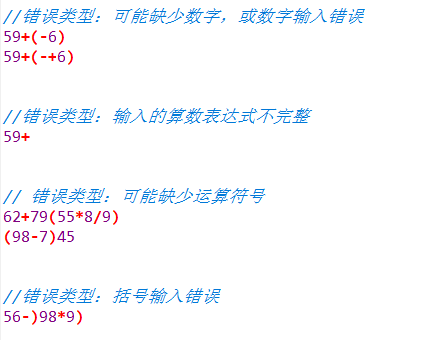
括号输入错误

**如下图，错误处理程序会在不同的情况下输出这4类错误信息，用于提示用户，经过测试，在大部分情况下，错误提示能够正确的提示出用户输入错在了哪里，可以作为一定的参考依据。**

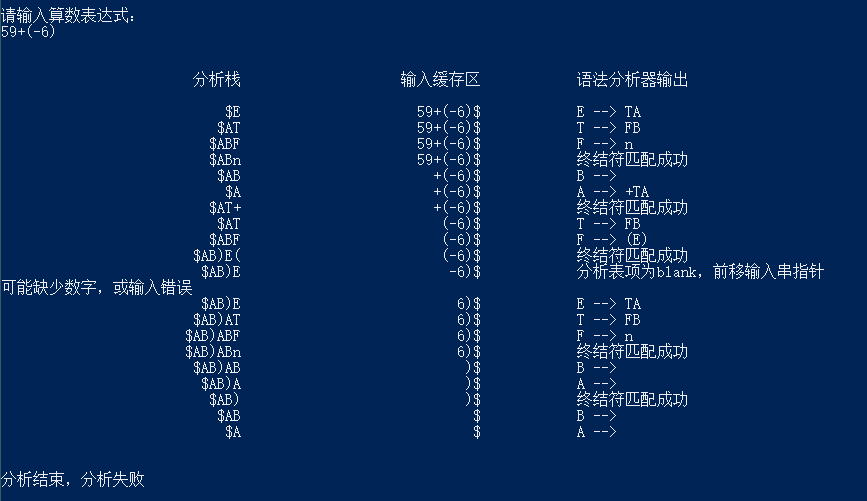
****

**下面是错误处理的截图：**

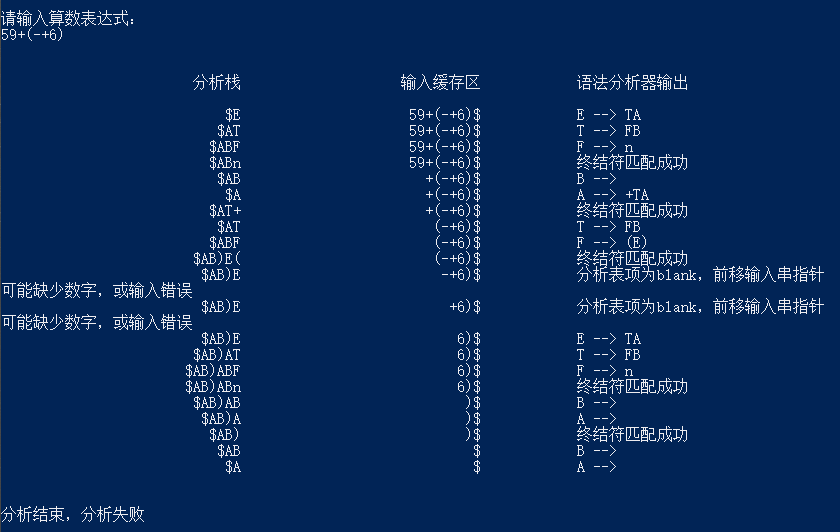
**样例：**

****

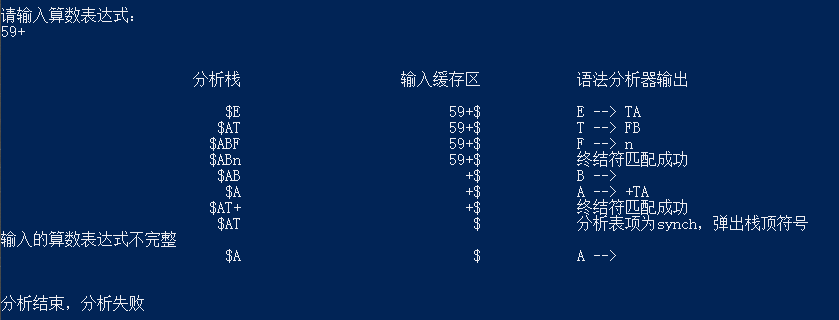
**59+(-6)：可能缺少数字，或数字输入错误**

****

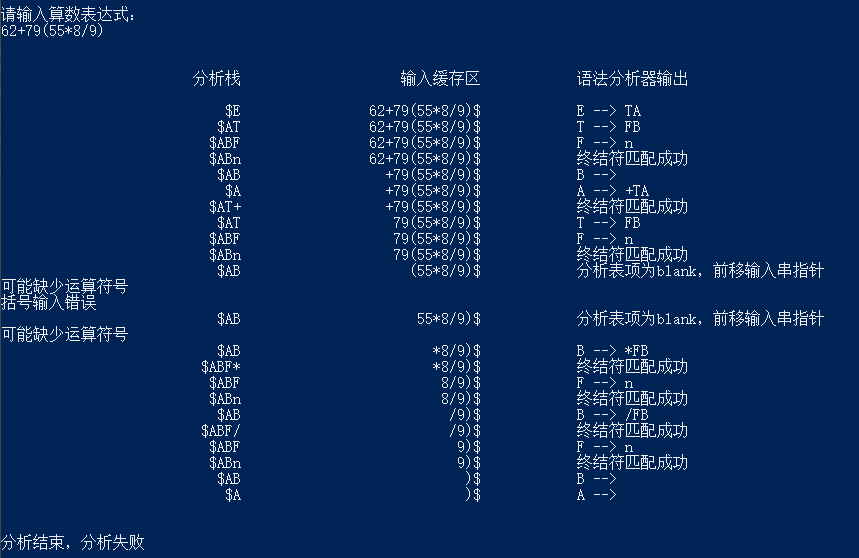
**59+(-+6)：可能缺少数字，或数字输入错误**

****

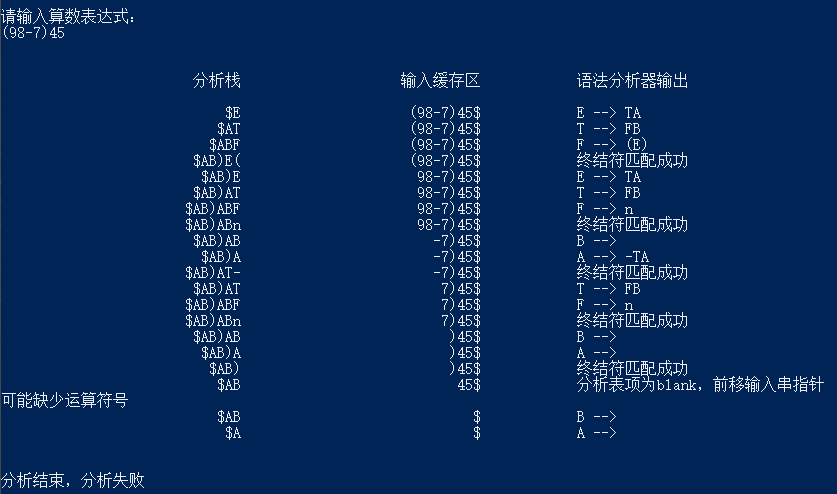
**59+：输入的算数表达式不完整**

****

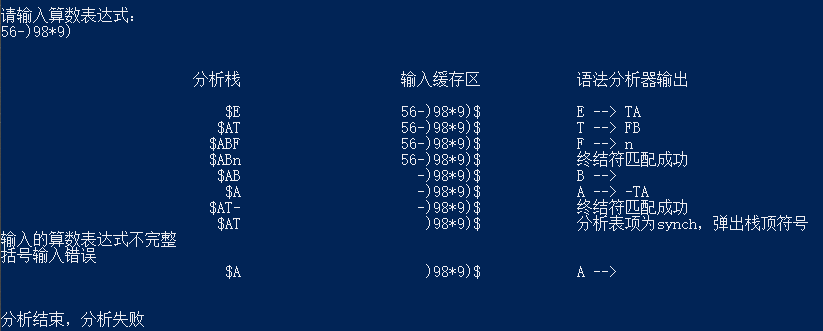
**62+79(55\*8/9)：可能缺少运算符号**

****

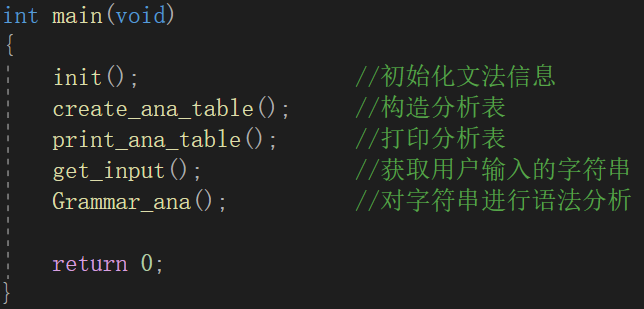
**(98-7)45：可能缺少运算符号**

****

**56-)98\*9)：括号输入错误**

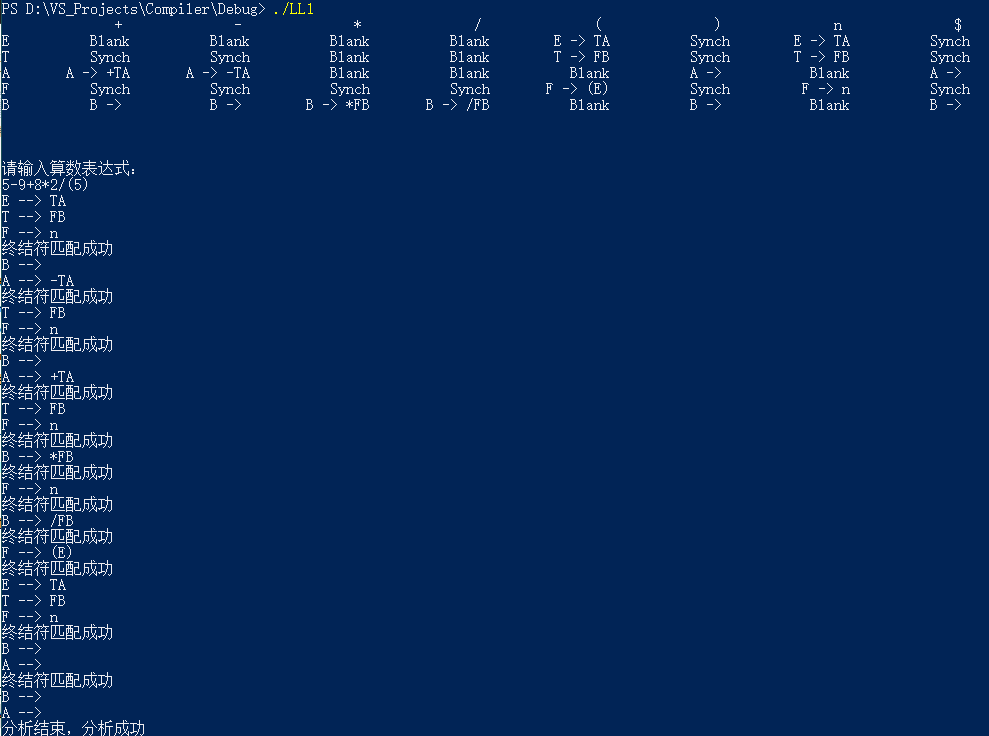
****

1. **程序结构**

****

1. **使用说明**
2. **执行方式：**

**建议使用命令行模式执行该程序：**



1. **打印详细分析过程**

**本程序设置了打印详细分析过程的模式，用户在命令行执行该程序可执行文件时，可以输入 –d，以打印详细信息，如图：**

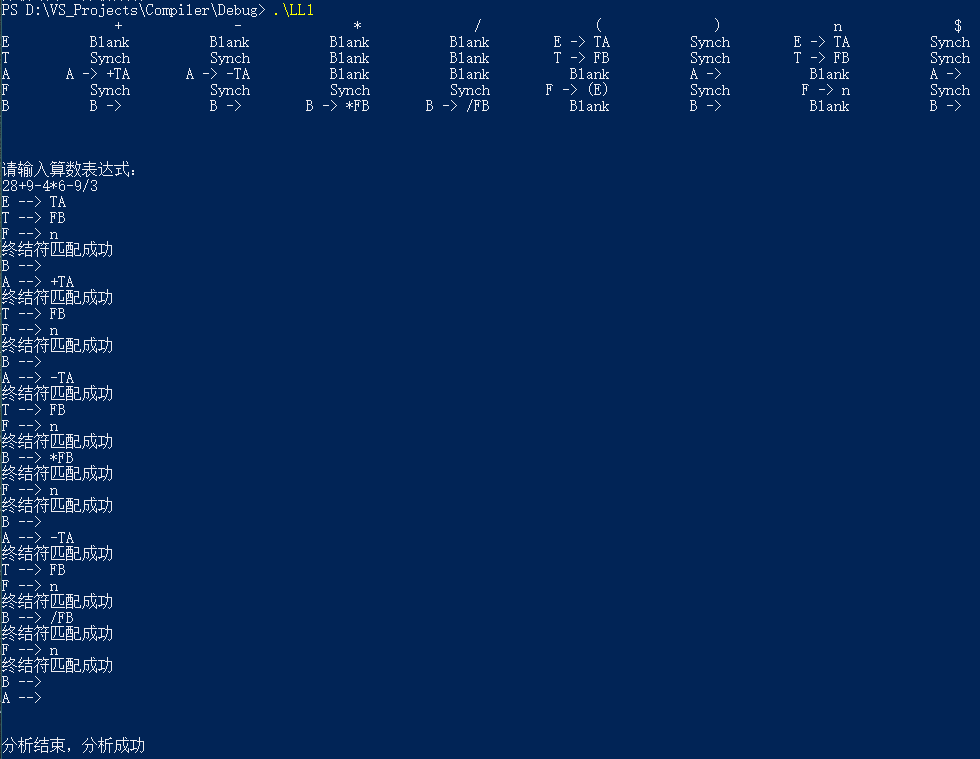
**注意：下图这种方式能够详细的输出分析栈和输入缓冲区的实时信息，但是由于打印格式的原因，仅支持输入运算表达式长度为30个字符以内，过长的字符串会导致输出异常（格式上的问题）。**

**如果要测试超长的运算表达式请勿选择这一模式！**

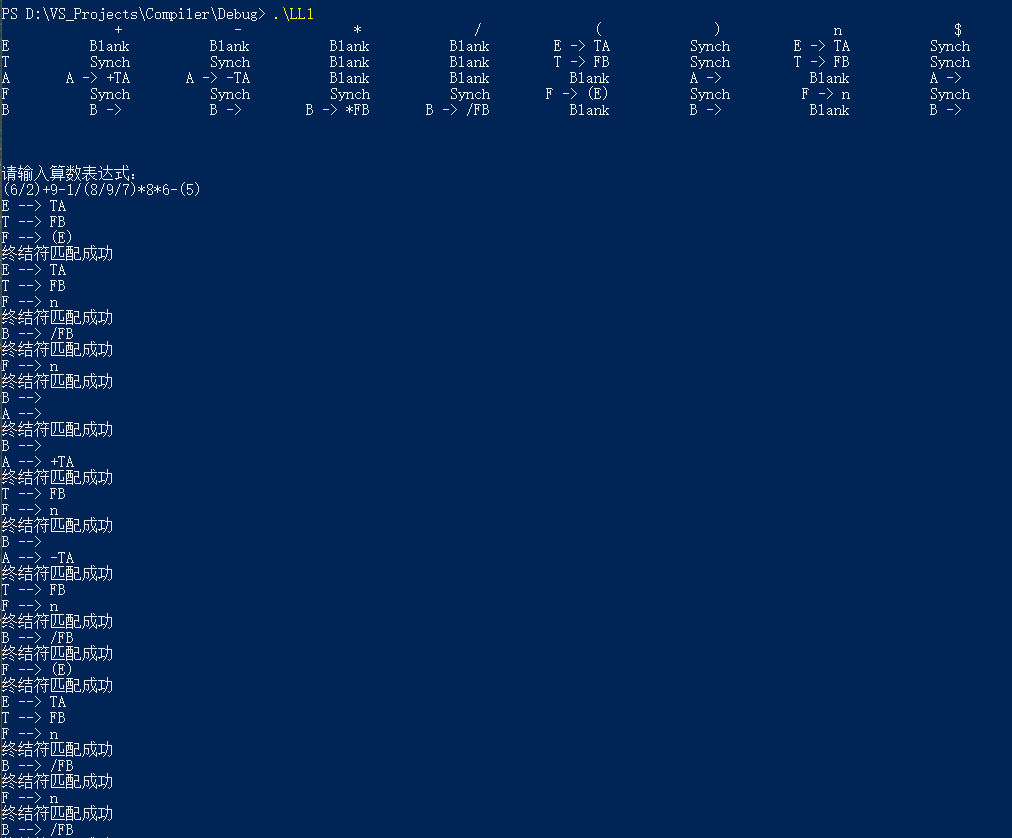
****

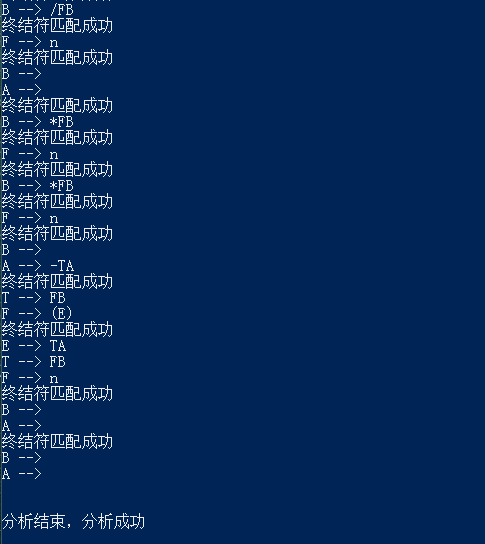
****

1. **运行截图**
2. **普通模式：28+9-4\*6-9/3**

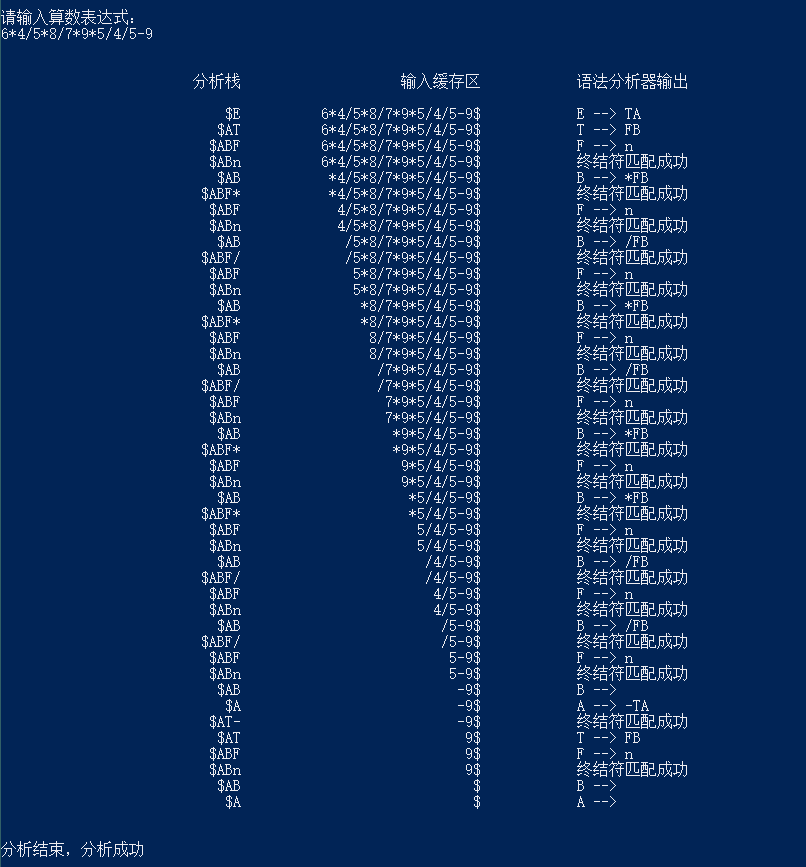


1. **普通模式：(6/2)+9-1/(8/9/7)\*8\*6-(5)**

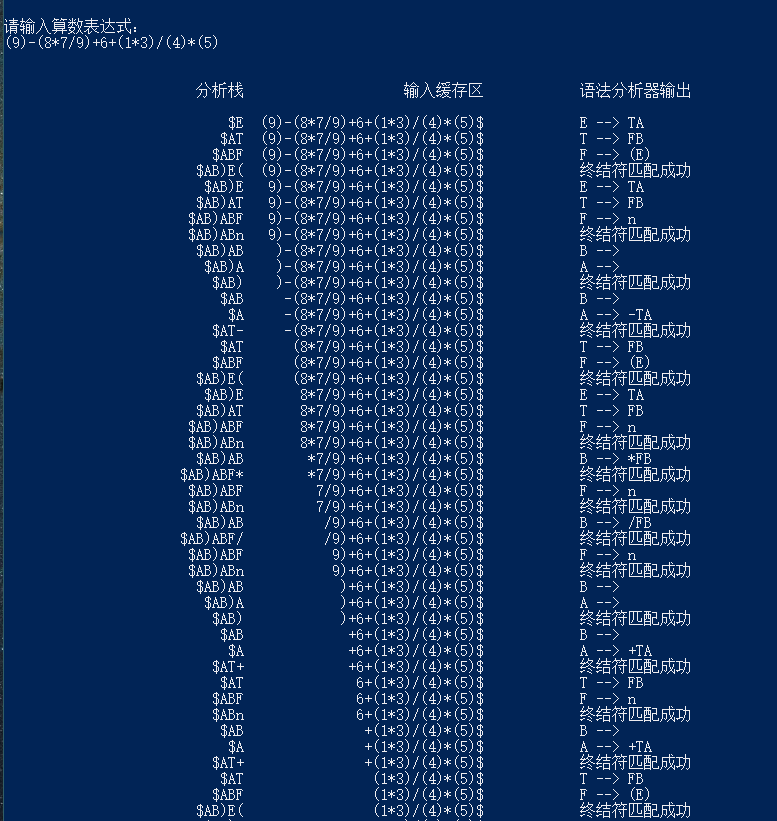


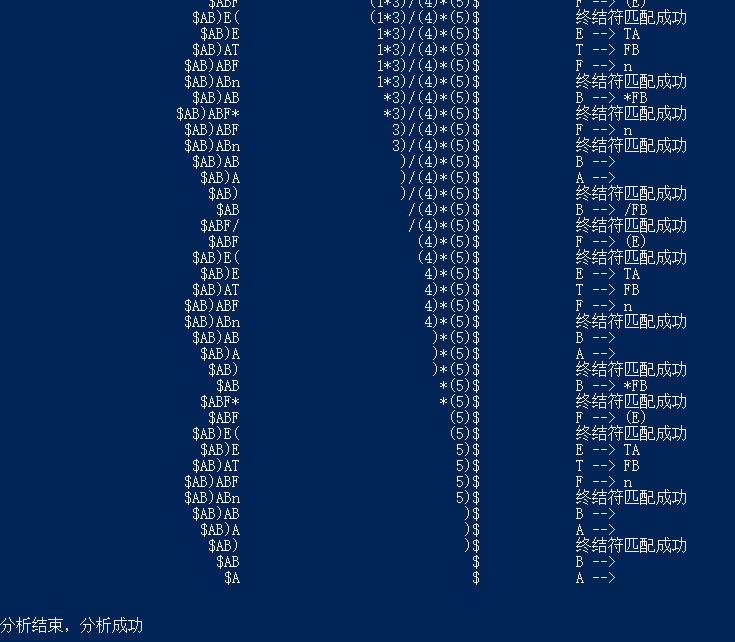


1. **详细分析过程模式：6\*4/5\*8/7\*9\*5/4/5-9**

****

1. **详细分析过程模式：(9)-(8\*7/9)+6+(1\*3)/(4)\*(5)**

****

****

1. **详细分析过程模式：(9\*8/563)(89-5+7)/**

****